



## ***An Open Source & Evergreen Glossary***

### ***The Cathedral & the Bazaar***

by Eric S. Raymond (O'Reilly, 2001). A must read if you are new to open source. A must read if you are not. A must read because it is the first great book to come out of the computer revolution.

Librarians who read it and hear the poetry behind the details will realize that open source works very much like we do.

### ***community***

In the **open source** world, there is much talk about the community of users and developers. They communicate with each other and, in fact, the developers are usually users. One of **Equinox's** missions is to foster the community that is growing up around **Evergreen**. Effectively the software and its development course is directed by the community. The community's home is at the **Evergreen** Website (<http://www.open-ils.org/>).

### ***compiled***

Describes a program written in computer language. It is usually not human readable. **Proprietary** software in the library world will be compiled.

### ***Equinox Software, Inc.***

The company founded by the developers of **Evergreen**. It supports and extends the capabilities of Evergreen and works to foster the community that has evolved to work on the Evergreen project. It also supports libraries implementing Evergreen. Its Website at: <http://www.esilibrary.com/>

### ***Evergreen***

The first consortially viable ILS, and the **open source** software that runs the **PINES** public library network. Its Website is at: <http://www.open-ils.org/>

### ***FOSS or FLOSS***

Free (Libre) **Open Source** Software

### ***FUD***

Fear, Uncertainty, Doubt. Do you want to trust your library to open source software written by a bunch of dope smoking hippies with orange hair?

*FulfilLment (tm)*

A development effort by **Equinox Software**. Its purpose is to create a virtual PINES-like consortium by using the proven **Evergreen** back end processing and opportunistic connectors to other ILS products. It would provide real-time discovery across a distributed network.

*GPL*

Gnu General Public License. An **open source** license that is used by **Evergreen** and most open source applications. There are various versions of the GPL and other kinds of open source licenses. Wikipedia's article:  
<http://en.wikipedia.org/wiki/GPL>

*GPLS*

Georgia Public Library Service, the state library of Georgia. GPLS administers the **PINES** network. Its Website is at: <http://www.georgialibraries.org/>

*migration*

If you change ILS vendors, you will have to move your data from one vendors' database structures to another's. Patron, transaction, and bibliographic records will have to be moved. This is normally not a process undertaken lightly. If the data are in a **proprietary** database, do you own your data so you can migrate?

*OS*

**Open source** software

*OpenSRF*

**Open Service Request Framework**, pronounced open surf. The software architecture at the core of the Evergreen ILS and the **FulfilLment** consortial borrowing platform. Invented by the developers of Evergreen, OpenSRF provides transparent load balancing, high-availability and abstraction features to applications, allowing developers to focus on functionality instead of infrastructure.

*open source*

Open source is a number of things. It is a class of licenses, a culture, a community, and a way of producing and sharing software.

In these senses, it is normally distinguished from **proprietary** licenses or software. Software produced by this method is released under an open source license like the **GPL** and the **source code** is freely available. Open source permits

*open source (continued)*

users to adapt, make changes, and improve software as long as the adapted software is also released through an open source license.

Open source is relatively new to the library world. One normally speaks of the alternative **proprietary** vendors as **legacy** or **traditional** vendors.

*open source software advantages:*

If there is a company such as **Equinox** to support an **open source** application, such applications have about the same as with **proprietary software advantages**.

Easy customization for your own local situation

Fast development - release early, release often

Cost it's free.

*open source software disadvantages:*

Who supports it if you can't?

It's free but it may not be cheap

*PINES*

The Georgia statewide public library resource sharing network. It currently has about 48 systems, 275 libraries, and circulates about 17 million items a year.

**Evergreen** software runs PINES.

*PostgreSQL*

PostgreSQL, sometimes shortened to Postgres, is a powerful, open source relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. To learn more about PostgreSQL visit <http://www.postgresql.org/>.

*proprietary*

A method for producing software that is normally distinguished from **open source** software. Proprietary software is not normally distributed as **source code** but as compiled programs so that one cannot see what the code does. It would normally be only supported by the company that manufactured it which can lead to **vendor lockin**.. Since users cannot see the code, they cannot make improvements or changes in it and have to wait for the next release..

*proprietary software advantages:*

Supported by the company you buy it from (can be good)

Normally **turnkey**

Support and documentation is said to be better than open source

You don't have to worry your pretty little head about your software.

*proprietary software disadvantages:*

Supported by the company you buy it from (can be bad if it is the only option because of potential **vendor lockin.**)

Slow development cycle

Some vendors have been bought by firms that don't know the industry and support and development have suffered

*Service Oriented Architecture*

A software architecture based on a collection of loosely-coupled, distributed services which communicate and interoperate via agreed standards. OpenSRF is an example of Service Oriented Architecture.

*source code*

...any sequence of statements and/or declarations written in some human-readable computer programming language. (Wikipedia). This is the code as it is written by the developers. Before it can be run on computers, it must be **compiled** into language that these computers can read..

*"Turnkey"*

In software, an application or suite of applications that a vendor sets up and all you have to do is turn the key and you are in business. More often heard of than actually seen in the real world, however.

*Vaporware*

Software that does not exist...but has been promised.

*Vendor lockin*

If you buy from a proprietary vendor, they are protected from competition for your business by 1) your multi-year contract, and 2) from the horrors of data **migration.**